



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **57069344 A**(43) Date of publication of application: **28 . 04 . 82**

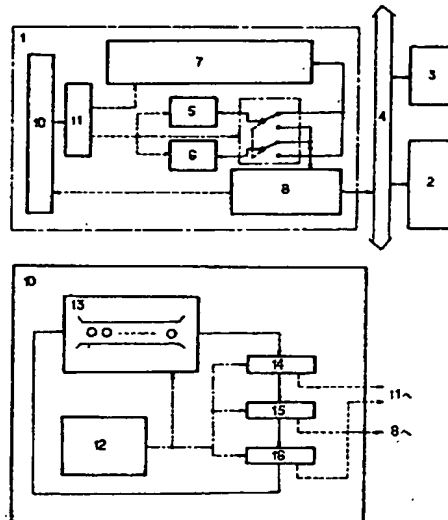
(51) Int. Cl

G06F 9/46(21) Application number: **55142650**(22) Date of filing: **13 . 10 . 80**(71) Applicant: **NIPPON TELEGR & TELEPH
CORP <NTT>**(72) Inventor: **IMASE MAKOTO
YOSHIDA YASUYUKI
ICHIKAWA HARUHISA****(54) CONTROLLER OF STORED PROGRAM****(57) Abstract:**

PURPOSE: To eliminate a complicated scheduler program and to delete the time overhead of a process switching operation, by incorporating a scheduler process switching hardware mechanism into a controller.

CONSTITUTION: A memory 2 and an input/output device 3 are connected to a controller 1 via a bus 4 to form a stored program controller. A scheduler 10 is provided within the controller 1 to switch and control a switch mechanism 11. Furthermore a scheduler control circuit 12, a process number memory 13 that has a cue structure and stores the process number within the controller 1, pre- and post-process registers 14 and 16 that control the mechanism 11 a process register 15 that stores and then transfers the process number which is being executed at an arithmetic device 8 are provided to the scheduler 10. Thus a complicated scheduler program is eliminated, and at the same time the time overhead is avoided for a process switching operation.

COPYRIGHT: (C)1982,JPO&Japio



⑬ 日本国特許庁 (JP)

⑪ 特許出願公開

⑫ 公開特許公報 (A)

昭57—69344

⑤ Int. Cl.³
G 06 F 9/46

識別記号

庁内整理番号
6745—5B

⑬ 公開 昭和57年(1982)4月28日

発明の数 1
審査請求 未請求

(全 5 頁)

⑭ 蓄積プログラム制御装置

⑯ 特 願 昭55—142650

⑰ 出 願 昭55(1980)10月13日

特許法第30条第1項適用 昭和55年9月11日
発行電子通信学会通信部門全国大会講演論文
集に発表

⑱ 発 明 者 今瀬真

武蔵野市緑町3丁目9番11号日
本電信電話公社武蔵野電気通信
研究所内

⑲ 発 明 者 吉田靖之

武蔵野市緑町3丁目9番11号日
本電信電話公社武蔵野電気通信
研究所内

⑳ 発 明 者 市川晴久

武蔵野市緑町3丁目9番11号日
本電信電話公社武蔵野電気通信
研究所内

㉑ 出 願 人 日本電信電話公社

㉒ 代 理 人 弁理士 鈴木誠

明 細 書

発明の名称

蓄積プログラム制御装置

特許請求の範囲

1. 実時間処理を必要とする複数のプロセスを並列、多重に実行する蓄積プログラム制御装置において、各プロセスのプログラムカウンタの値、プログラム状態レジスタの内容、ベースレジスタの内容、汎用レジスタの内容などの各プロセス固有の情報を記憶するメモリと、所望のタイミングで次々に実行すべきプロセスを決定していくスケジューラと、前記スケジューラの指示のもとづいて前記メモリと演算装置の間で前記各プロセス固有の情報を順次転送してプロセスの切り換えを行うプロセス切り換え機構とを有することを特徴とする蓄積プログラム制御装置。

発明の詳細な説明

本発明は複数のプロセスを並列、多重に実行する蓄積プログラム制御装置に関し、詳しくは、そのプロセス切り換えのスケジューリング方式及び

プロセス切り換えのハードウェア機構に関するものである。

従来の蓄積プログラム制御装置は、一般に1つのプログラムカウンタをもち、プログラムカウンタの値を変更することによりプログラムシーケンスを制御しており、1つのコントロールフローをなわち1つのプロセスのみを持つ事を基本としていた。複数のプロセスを1つの制御装置で並列、多重に動作させるマルチプロセス処理の機能は、スケジューラなどと呼ばれるプログラムにより実現されていた。通常のマルチプロセス処理では、入出力装置、タイマの割込みなどにより1つのプロセスを中断し、制御はこのスケジューラに移る。スケジューラは、実行を中断した該プロセスの固有情報を持つレジスタ類の内容をメモリに退避するなどして該プロセスを実行待ち状態とし、割込み原因、入出力装置の状態、他のプロセスの状態などシステム全体の状態を調べ、次に実行させるプロセスを決定し、あらかじめ退避されていた該プロセスの固有情報をレジスタ類に復帰するなど

して該プロセスを実行可能状態にし、該プロセスに制御を移す。この一連の動作の繰り返しにより複数プロセスを並列、多重に実行させることを実現している。

しかし、このような方式では、多くの場合、割込みレベルが多重であることや、システム全体の状態を把握する必要があることから、スケジューラは複雑なプログラム構造となる欠点があつた。また、プロセス切り換えが頻繁に行われるシステムでは、プロセス切り換え処理の時間的オーバーヘッドが大きく、制御装置の処理能力が低減する欠点があつた。

特に、電子交換機のように、数百から数千という多数の呼を各々独立したプロセスとして並列して制御する蓄積プログラム制御装置上のプログラムでは、この欠点を補うために、1つの呼から他の呼の制御に移るプロセス切り換え時に、各プロセスで本来そのプロセス自体の処理とは独立と考えられるスケジューリング方式及び他のプロセスの状態を考慮して、時間的オーバーヘッドを減少さ

せる工夫を行つてゐるが、処理呼数が増えた場合にはその効果には限界がある。逆に、この方策を施したため各プロセスを独立したプログラムとみなすことができず、プログラムの理解が困難となり、保守、運用、維持には高度な知識が要求されるという問題が生じている。

また、交換処理のように、入出力装置の制御が大きな部分を占めているプロセス制御においては、機能の追加・変更がある時、入出力装置及びその信号方式の追加・変更が必要となる。入出力装置の制御の一部はスケジューラプログラムに含まれているため、このような場合、先に説明したような複雑な構造のスケジューラプログラムの変更が必要であり、機能の追加・変更が困難である。

このように、従来方式においては、多重プロセスの切替え制御を全てプログラムで行つていたため、プロセス切り換えの時間的オーバーヘッドにより制御装置の処理能力が低下すること、及びプロセス別の入出力装置の制御にもスケジューラプログラムが介在していたため、スケジューラプログ

ラムの構造は複雑であり、入出力装置の構成、操作法に依存して変わり、プログラムの生産性、融通性、保守性が悪い等の欠点があつた。

本発明はこのような従来の欠点を除去するため、入出力装置などからの割込みをきつかけにプロセス切り換えを行なわず、一命令(あるいは一定時間単位)ごとにプロセスを順次切り換えるスケジューリング方式をとり、かつ、スケジューリング機能と入出力制御機能を分離し、スケジューリング機能を制御装置に内蔵して、複数プロセスを1つの制御装置で動作させるにあたり、プロセス切り換えの時間的オーバーヘッドがなく、スケジューラプログラムが不要な制御装置を構成したもので、以下図面について詳細に説明する。

第1図は本発明の実施例のブロック図である。図において、1は本発明にかゝる制御装置(プロセッサ)、2はプログラム、データを格納するメモリ、3は入出力装置、4は制御装置1とメモリ2、入出力装置3間のデータ転送を行うバスである。制御装置1の構成中、5、6は各々別々に

実行中、実行中断処理中、または実行準備中のプロセスの個有情報すなわち各プロセス別に対応したプログラムカウンタの値、プログラム状態レジスタの内容、汎用レジスタの内容などを格納する同一形式の内部レジスタ、7はすべてのプロセスの個有情報を格納する内部メモリ、8はメモリ2に格納されている命令をフェッチし実行する演算装置、9は内部レジスタ5、6と内部メモリ7、演算装置8の間の接続関係を切り換えるスイッチである。10はすべてのプロセスの中から次に実行するプロセスを決定するスケジューラ、11は切り換えスイッチ9の制御及び内部メモリ7と内部レジスタ5、6間の情報転送を行ないプロセス切り換えを行なう切り換え機構である。

制御装置1は次の6ステップの実行により機械語1命令を実行する。

(ステップ1) スケジューラ10は次に実行するプロセスを決定し、切り換え機構11にそのプロセス番号を通知する。

(ステップ2) 切り換え機構11は、次に実行する

プロセスの個有情報を内部メモリ7から内部レジスタ5(または6)に転送する。

(ステップ3) 切り換え機構11は、切り換えスイッチ9を切り換え、内部レジスタ5(または6)を演算装置8と接続する。このとき他の内部レジスタ6(または5)は内部メモリ7に接続される。(ステップ4) 演算装置8は、内部レジスタ5(または6)内のプログラムカウンタの指示するメモリ2上の命令をフェッチし、解説、実行する。(ステップ5) 切り換え機構11は、切り換えスイッチ9を切り換え、内部レジスタ5(または6)を内部メモリ7に接続する。このとき、他の内部レジスタ6(または5)は演算装置8に接続される。

(ステップ6) 切り換え機構11は、内部レジスタ5の内容を内部メモリ7に転送する。

第2図は、例として3つのプロセス①、②、③が並列、多重に動作する時、(ステップ1)から(ステップ6)の実行関係をタイミングチャートで示したものである。各タイミング[1]、[2]、...

[5]の時間は、演算装置8が(ステップ4)の動作を行う時間に相当する。第2図において、例えばプロセス②はタイミング[2]で(ステップ1)、(ステップ2)、タイミング[3]で(ステップ4)、タイミング[4]で(ステップ6)が実行される。つまり、この例ではプロセス切り換えを1命令ごとに行うが、プロセス切り換え単位は1命令ではなく複数命令あるいは一定時間単位などにもすることも同様の原理により可能である。

(ステップ1)ではスケジューラ10、(ステップ2)と(ステップ6)では内部レジスタ5(または6)と内部メモリ7、(ステップ4)では演算装置8と内部レジスタ5(または6)が動作する。切り換え機構11は、切り換えスイッチ9を動作させ、内部レジスタ5、6の一方を内部メモリ7、他方を演算装置8にそれぞれ接続し、これをプロセス切り換え時間単位ごとに交互に切り換える。このように(ステップ1)から(ステップ6)の処理を、いわゆるパイプライン方式で処理しているため、演算装置8の実行時間には空き時間がな

く、プロセス切り換え処理の時間的オーバーヘッドはない。

スケジューラ10は、制御装置1内に存在するすべてのプロセスを順次切り換えるスケジューリング方式をとる。第3図は、このスケジューリング方式を実現するスケジューラ10の一実施例である。12はスケジューラ10の制御をつかさどるスケジューラ制御回路、13は制御装置1内に存在するプロセス番号を記憶するキュー構造をしたプロセス番号メモリ、14は内部メモリ7から内部レジスタ5または6にプロセス個有情報を転送中のプロセスの番号を格納する前処理プロセスレジスタ、15は演算装置で実行中のプロセスの番号を格納する処理プロセスレジスタ、16は内部レジスタ5または6から内部メモリ7へプロセス個有情報を転送中のプロセスの番号を格納する後処理プロセスレジスタである。スケジューラ制御回路12は、プロセス切り換え時刻すなわち第2図のタイミング[1]、[2]、.....の切れ目を判断し、その時刻に処理プロセスレジスタ15に記憶されている番号を後処理プ

ロセスレジスタ16に転送し、更に前処理プロセスレジスタ14に記憶されているプロセス番号を処理プロセスレジスタ15に転送し、後処理プロセスレジスタ16のプロセス番号をプロセス切り換え機構11に、処理プロセスレジスタ15のプロセス番号を演算装置8に通知する。その後、プロセス番号メモリ13のキューの先頭のプロセス番号を読み出して前処理プロセスレジスタ14に転送し、このプロセス番号をプロセス切り換え機構11に通知する。また、後処理プロセスレジスタ16に記憶されているプロセス番号をプロセス番号メモリ13のキューの最後尾に記憶させる。この動作を各タイミング毎に行うことにより、プロセスを順次切り換えることができる。

これにより各プロセスは、見かけ上実行速度の遅い演算装置を専有して使用しているように見える。第2図の例では、各プロセス①、②、③は3タイミング中に1タイミングずつ演算装置8を専有して処理が進行するので、各プロセスは演算装置8の1/3の実行速度を持つ演算装置を専有して

いるように見える。各プロセスでは、入出力装置を直接アクセスして入出力装置の動作終了、外部からの信号待ちを判断し、入出力装置を制御することが可能である。すなわち、割込み機能がなくとも入出力装置を制御することが可能となり、割込み種別、入出力装置の状態などシステム全体の状態を一括して管理していた従来のスケジューラプログラムが不要となる。

また、各プロセスの見かけ上の実行速度を何種類か用意し、見かけ上の実行速度を変化させる命令を設けることも次のようにして可能である。第4図は一例として10 msecに1命令ずつ実行するプロセス群と、任意の速度で実行してもよいプロセス群の2つの見かけ上の実行速度をもつ場合のスケジューラ10の一実施例である。第3図の実施例と異なる点は、プロセス番号メモリ13が、任意の速度で実行しているプロセス番号を格納するメモリ17と10 msec/一命令で実行されているプロセス番号を格納するメモリ18の2つのキューメモリを持つこと、及び10msecに一度ずつスケジューラ制

御回路12へプロセス番号を送るタイマー19があることである。

(ステップ2) 特別のプロセス番号を取り出した後は、次にタイマー19より10 msec毎に送られてくる信号を受け取るまでキューメモリ17より前処理レジスタ14へプロセス番号を送送し、(ステップ1)へ戻る。

この機能を用いることにより、各プロセスのプログラムでは、入出力装置の実行終了待ち状態では10 msec/一命令の実行速度で入出力装置の状態を眺めとり実行を終了したかを判定し、終了後は実行速度を任意に切り換え内部処理を行うことが可能になる。これにより、入出力装置の実行終了、外部からの信号などを待つ状態にあるプロセスが演算装置を専有する比率を減少させることができ、1台の制御装置で実行できるプロセス数を増やすことができる。

また、電子交換機のように各プロセスの処理内容が同一の場合、各プロセスの状態はプロセス対応に存在するプログラムカウンタ、レジスタ類に保持し、プログラムをリエントラント構造にすれば、複数プロセスで1つのプログラムを共有させることも可能である。

以上説明したように、本発明によれば、スケジューラ、プロセス切り換えハードウェア機構を制御装置内に内蔵しているために、複雑な構造を持つスケジューラプログラムが不要となる利点がある。また、プロセス切り換え処理の時間的オーバーヘッドがなくなり、制御装置の処理能力が向上の利点がある。特に、変換システムに適用すれば、入出力装置の細かい制御を行う部分までが各プロセスのプログラムに含まれ、各プロセスのプログラムを独立とみなせるため、プログラムが理解しやすく、プログラムの保守、運用、維持が容易となり、機能追加・変更が簡単になるという利点がある。

図面の簡単な説明
第1図は本発明にかゝる制御装置の一実施例を示す図、第2図は第1図の動作例を説明するためのタイミング図、第3図及び第4図は第1図のスケジューラの一実施例を示す図である。

図面は、タイマー19より10 msec毎に送られてくる信号を受け取ると、時刻の切れ目を示すための特別のプロセス番号をキューメモリ18に書き込む。その後、特別のプロセス番号がくるまでキューメモリ18より、前

御回路12へプロセス番号を送るタイマー19があることである。その動作で異なる点は、スケジューラ制御回路12がプロセス番号メモリ13と前処理レジスタ14、後処理レジスタ16間で、プロセス番号を送送制御する際、プロセス番号メモリ13内のキューメモリ18と19のどちらに転送するかを選択することである。この選択の方法について以下で述べる。

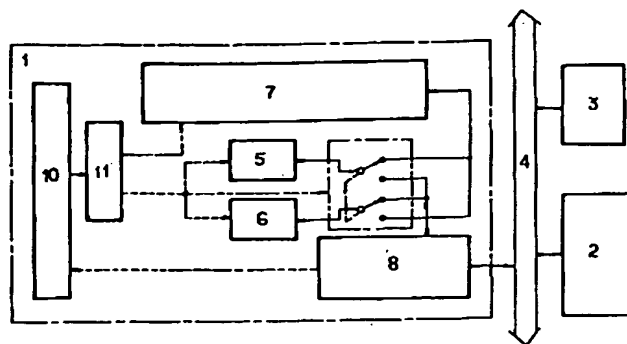
後処理レジスタ16からプロセス番号メモリ13へプロセス番号を送送する場合、通常はそのプロセス番号が取り出されたキューメモリにつなぐ。もし、演算装置8がそのプロセスの実行速度を変更する命令を検出した時は、その要求に従ったキューメモリに接続する。プロセス番号メモリ13より前処理レジスタ14へプロセス番号を送送する場合、キュー選択は次のように行う。

(ステップ1) スケジューラ制御回路12は、タイ

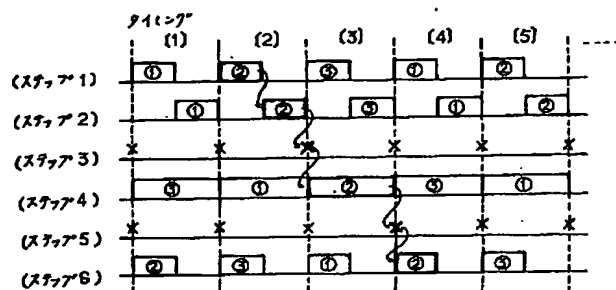
マ19より10 msec毎に送られてくる信号を受け取ると、時刻の切れ目を示すための特別のプロセス番号をキューメモリ18に書き込む。その後、特別のプロセス番号がくるまでキューメモリ18より、前

1…制御装置、 2…メモリ、 3…入出力装置、 4…バス、 5、6…内部レジスタ、 7…内部メモリ、 8…演算装置、 9…切り換えスイッチ、 10…スケジューラ、 11…切り換え機構、 12…スケジューラ制御回路、 13…プロセス番号メモリ、 14…前処理プロセスレジスタ、 15…処理プロセスレジスタ、 16…後処理プロセスレジスタ、 17、18…キューメモリ、 19…タイマ。

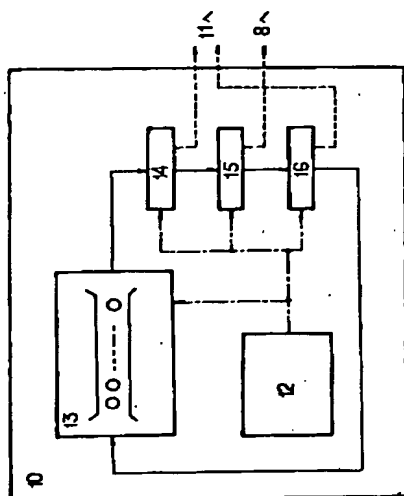
第 1 図



第 2 図



第 3 図



第 4 図

